Cai et al. BMC Genomics (2023) 24:52
https://doi.org/10.1186/s12864-022-09034-1

BMC Genomics

RESEARCH

Open Access

GM-lncLoc: LncRNAs subcellular localization prediction based on graph neural network with meta-learning

Junzhe Cai[1], Ting Wang[1], Xi Deng[1], Lin Tang[2] and Lin Liu[1*]

Abstract

In recent years, a large number of studies have shown that the subcellular localization of long non-coding RNAs (lncRNAs) can bring crucial information to the recognition of lncRNAs function. Therefore, it is of great significance to establish a computational method to accurately predict the subcellular localization of lncRNA. Previous prediction models are based on low-level sequences information and are troubled by the few samples problem. In this study, we propose a new prediction model, GM-lncLoc, which is based on the initial information extracted from the lncRNA sequence, and also combines the graph structure information to extract high level features of lncRNA. In addition, the training mode of meta-learning is introduced to obtain meta-parameters by training a series of tasks. With the meta-parameters, the final parameters of other similar tasks can be learned quickly, so as to solve the problem of few samples in lncRNA subcellular localization. Compared with the previous methods, GM-lncLoc achieved the best results with an accuracy of 93.4 and 94.2% in the benchmark datasets of 5 and 4 subcellular compartments, respectively. Furthermore, the prediction performance of GM-lncLoc was also better on the independent dataset. It shows the effectiveness and great potential of our proposed method for lncRNA subcellular localization prediction. The datasets and source code are freely available at https://github.com/JunzheCai/GM-lncLoc.

Keywords: lncRNA, Subcellular localization, Graph neural network, Meta-learning, Classification

Introduction

The RNAs that cannot encode proteins are called non-coding RNA (ncRNA) [1], which can be further divided into two categories according to their molecular chain length: small non-coding RNA (sncRNA) with molecular chain length less than 200 nucleotides and long non-coding RNA (lncRNA) with molecular chain length more than 200 nucleotides [2]. In the past, lncRNAs were initially considered as the "noise" of genome transcription, which was the by-product of RNA polymerase II transcription and had no biological function [3]. However, more and more studies have shown that lncRNAs are involved in many biological functions. Moreover, abnormal behavior of lncRNAs leads to the formation of several types of cancer, Alzheimers disease, Huntingtons disease, and cardiovascular diseases [4–13]. Obviously, a better understanding of lncRNA function would enhance our understanding of specific cell development and physiology. Several studies have shown that the function of lncRNA is highly dependent on its position inside the cell [14–16]. Therefore, identification of lncRNA subcellular localization is particularly important.

There are two main types of methods for predicting lncRNA subcellular localization. One is biochemical experiments, which have the advantage of precise positioning results and have the disadvantage of being time-consuming and expensive. Therefore, more and more

*Correspondence: liulinrachel@163.com

[1] School of Information, Yunnan Normal University, Kunming, Yunnan, China
Full list of author information is available at the end of the article

researchers have tried to find a breakthrough in computational methods, which have the advantages of being time-saving, efficient and stable. Especially with the solid foundation provided by lncRNA subcellular localization databases, including RNALocate [17], LncATLAS [18], and lncSLdb [19], computational model-based lncRNA subcellular localization methods have become a new trend in this field of research.

At present, several computational models have been used to predict the subcellular localization of proteins with high accuracy [20–24]. Such problem of protein or RNA subcellular localization prediction is essentially a classification process in machine learning. Therefore, the current studies also follow the general process of classification prediction, including dataset building, lncRNA feature extraction, and classifier training. Zhen C, et al. [25] proposed the lncLocator method, which utilizes support vector machines (SVM), Random Forest (RF) and neural network (NN) to predict subcellular localizations of lncRNAs and yield an overall accuracy of 59.1% on benchmark dataset with 5 subcellular compartments; Gudenas, B.L., et al. [26] and Yang Lin, et al. [27]. developed the deep learning algorithm to predict subcellular location on a large dataset with 2 classes; Furthermore, several researchers focus on the benchmark dataset with 4 subcellular compartments. Generally, SVM algorithm is widely used as the classification model in predicting lncRNA subcellular localization, such as iLoc-lncRNA proposed by Su Z D, et al. [14], Locate-R proposed by Aa A, et al. [28] and Xiao-Fei Yang, et al. [29], which get an accuracy of 86.11, 90.69 and 92.38%, respectively; also for the benchmark dataset with 4 subcellular compartments, Fan Y, et al. [30] come up with a method based on logistic regression, lncLocPred, which obtains 92.37% accuracy.

Although these aforementioned methods have made some progress in lncRNA subcellular localization prediction, the prediction accuracy varies greatly due to the different label and sample numbers of datasets. Gudenas, B.L., et al. [26] and Yang Lin, et al. [27] utilize a large amount of data and less subcellular localization labels, so relatively high prediction accuracy is obtained. In the rest of studies [14, 25, 28–30], the dataset contains only a few hundred samples with 4 or 5 subcellular compartments, which belongs to the few-shot learning field. From the perspective of computational models, a small number of samples is a big obstacle to the training of classifier, which significantly limits the improvement of prediction accuracy. Especially for the deep learning methods, it is able to automatically capture advanced features of data, but it is not good at getting better generalization performance for few-shot learning. Therefore, for the dataset with 4 or 5 subcellular compartments, previous studies mainly made use of traditional machine learning methods to predict lncRNA subcellular localization and spent a lot of resources on feature extraction. For instance, Zhen C, et al. [25] specially used an unsupervised stacked autoencoder model to obtain high-level features from k-mer low-level features; Fan Y, et al. [30] utilized k-mer, PseDNC and TRIPLET methods to extract features, and then fused these features through a series of operations. Although some recent studies have tried to utilize deep learning to predict lncRNA subcellular localization on dataset with a few lncRNAs, they have obtained poor performance. As an example, the accuracy of DeepLncLoc proposed by Zeng M, et al. [31] is only 53.7% in the dataset with 5 subcellular compartments.

In view of above problems, this paper proposed a new prediction model called GM-lncLoc, which mainly explores how to predict lncRNA subcellular localization in a few samples dataset based on advanced lncRNA features automatically extracted by deep learning. On the one hand, GNN [32] is a powerful model that can aggregate the node features and the information of graph structure, which is conducive to the node classification task of lncRNA subcellular localization research. Therefore, after extracting the low-level features of lncRNA sequences by simple k-mer method, the hidden representation of lncRNA sequences is automatically captured as high-level features based on GNN in our model. On the other hand, meta-learning is an efficient way for dealing with few-shot learning that extracts meta-knowledge from multiple similar tasks, allowing the predictor to acquire the ability of other similar classification tasks quickly. In the field of meta-learning, there are many models that are widely accepted and considered effective, such as MAML [33] and Reptile [34]. Inspired by the study of Kexin Huang [35] et al., we attempted to combine GCN [36] and MAML [33] to address the poor performance of deep learning in few-shot lncRNA subcellular localization learning. Generally speaking, GM-lncLoc not only obtains efficient lncRNA subcellular localization prediction on a small number of lncRNA samples, but also learns the meta-parameters with strong generalization ability for rapid adaptation to similar unseen task.

To our best knowledge, we are the first to identify lncRNA subcellular localization based on GNN and few-shot learning method. In general, the steps of GM-lncLoc are as follows: (1) constructing benchmark dataset; (2) balancing samples; (3) constructing graph; (4) Model: GCN based on MAML; (5) performance evaluation. See the flow chart in Fig. 1.

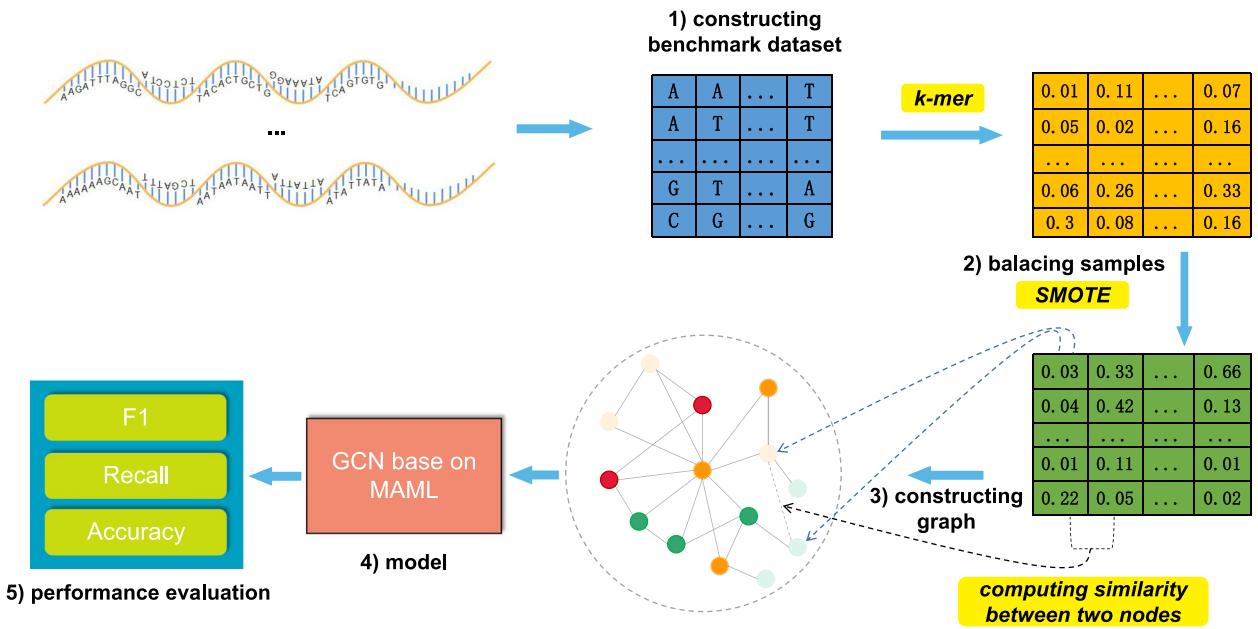

Fig. 1 The flow chart of GM-lncLoc: (1) constructing benchmark dataset; (2) balancing samples; (3) constructing graph; (4) predicting labels with the model (GCN based on MAML); (5) evaluating the model's performance with evaluation indicators

Table 1 Benchmark dataset

	Original1	After filtering	After SMOTE (dataset1)	Original2	After filtering	After SMOTE (dataset2)	dataset3
Cytoplasm	301	292	292	426	417	417	198
Nucleus	152	149	292	156	153	417	82
Cytosol	91	91	292	–	–	–	–
Ribosome	43	43	292	43	43	417	99
Exosome	25	25	292	30	30	417	16
Total	612	600	1460	655	643	1668	395

Materials and methods

Dataset

A high-quality dataset is crucial for effective and accurate prediction models, where the labels in the dataset are evenly distributed and have sufficient samples. As mentioned above, in the current studies of lncRNA subcellular localization prediction, researchers have mainly constructed three benchmark datasets: Zhen C, et al. [25] and Zeng M, et al. [31] constructed the 5 subcellular compartments dataset from RNALocate database; Gudenas, B.L., et al. [26] and Yang Lin, et.al [27]. constructed datasets with 2 subcellular compartments; other researchers have constructed datasets with 4 subcellular compartments. This section mainly introduces the construction of our two datasets, dataset1 and dataset2, which are based on the 5 subcellular compartments dataset of Zhen C, et al. [25] and the 4 subcellular compartments dataset of Su Z D, et al. [14], respectively. The steps of dataset construction are as follows:

Step 1: First, we download the raw data of Zhen C, et al. [25] and Su Z D, et al. [14] from the websites,[1] which contain 612 and 655 lncRNAs sequence, as shown in Table 1. After screening, to reduce information redundancy and noise interference, we removed 1 sequence of length 91,671 and 11 sequences containing special symbols "N, R, S and Y". Finally, dataset1 and dataset2 contain 600 and 643 lncRNAs sequences, respectively, including 292/417 Cytoplasm, 149/153 Nucleus, 91/– Cytosol, 43/43 Ribosome, 25/30 Exosome.

Step 2: Previous studies have shown that there are many factors related to lncRNA subcellular locali-

[1] http://www.csbio.sjtu.edu.cn/bioinf/lncLocator/ http://lin-group.cn/server/iLoc-LncRNA/download.php

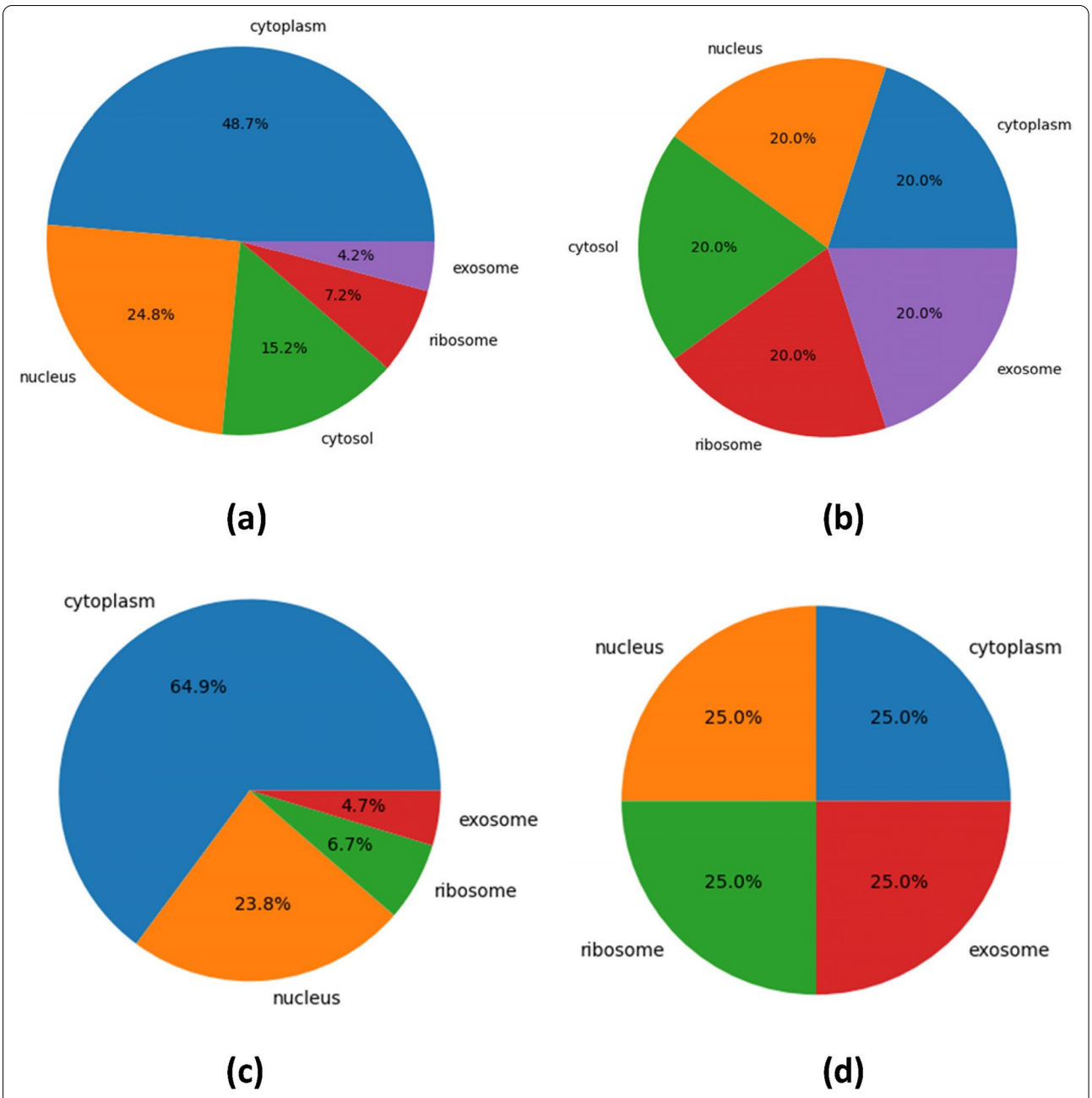

Fig. 2 (**a**): dataset1 before SMOTE; (**b**): dataset1 after SMOTE; (**c**): dataset2 before SMOTE; (**d**): dataset2 after SMOTE

zation, such as sequence and structure [37]. As it is still challenging to identify RNA structural information experimentally and theoretically [38], the approaches of current studies mainly extracted low-level features from lncRNA sequence [14, 25–31] based on k-mer [39], RevKmer [40, 41] and PseDNC [42–44] et al. K-mer can get the basic information of a sequence, and has a wide range of applications in many fields of bioinformatics [45–48]. In our experiment, the features of RNA sequences extracted by k-mer have been verified to be more effective than other feature extraction methods. Therefore, after extracting the low-level features of 600/643 lncRNAs sequences by k-mer, 600/643 vectors were obtained.

Step 3: As shown in Fig. 2 **(a)(c)**, the dataset is unbalanced with a few samples. At present, there are two main methods to balance samples: under-sampling

and over-sampling. The under-sampling method randomly selects the subsets of samples from each classification to consist of a balanced dataset [49, 50], which will led to loss of important information from original data.

However, the over-sampling method synthesizes new data for labels with only a few samples, which is more suitable for small and unbalanced dataset and it is also adopted in many other studies, such as lncLocator [25], Locate-R [28], and so on. Therefore, an over-sampling method Synthetic Minority Over-Sampling Technique (SMOTE) [51] is considered in this paper. Taking dataset1 as an example, SMOTE synthesizes the data as follows: (1) 292, the number of Cytoplasm classes with the largest number of samples in the original dataset, was chosen as a reference; (2) randomly select a sample in the Nucleus class as the central sample, and 143 nearest neighbors of this center sample were selected stochastically; (3) 143 samples are randomly generated along the line segments of the central sample and 143 nearest neighbors, and then the Nucleus class contains 292 samples, including 149 real and 143 synthetic samples; (4) the samples of Cytosol, Ribosome and Exosome were sampled according to (2) and (3), and finally, 292 samples were collected for each class. There are 1460/1668 samples in total in the final datasets after over-sampling. It can be seen in Table 1 and Fig. 2 **(b)(d)** that the distribution of labels in final datasets is balanced. However, the sample size is not enough to support the deep learning model to get good results.

Moreover, we prepare an independent test set, dataset3, provided by Fan Y, et al. [30].[2] We removed 1 sequence containing special symbols and got 395 samples, including 198 Cytoplasm, 82 Nucleus, 99 Ribosome and 16 Exosome.

Constructing graph

Constructing graph is a process of modifying the data format of low-level features into graphical data, which can be applied to GCN with the advantage of capturing structural information of the graph. In the field of bioinformatics, several researchers have constructed a protein sequence similarity network (SSN) [52–54] to study the properties of proteins. Correspondingly, the graph structure is constructed by cosine similarity of features in this paper. Meanwhile, GM-lncLoc is able to extract information from the perspective of non-Euclidean space, which is the most different from previous methods based on Euclidean space data. An appropriate graph structure facilitates GCNs to aggregate neighbor node information more efficiently.

[2] https://github.com/jademyC1221/lncLocPred/tree/master/lncLocPred/supplementary%20material

Problem Formulation

The graph is denoted as $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E},\boldsymbol{X})$, where $\boldsymbol{V}=\{\boldsymbol{v_1},\boldsymbol{v_2},\ldots,\boldsymbol{v_n}\}$ represents the node-set, $\boldsymbol{v_i}$ represents the i-th lncRNA sequence, which is one of the nodes in the graph $\boldsymbol{G}$. $\boldsymbol{E}=\{\boldsymbol{e_{1,2}},\boldsymbol{e_{1,3}},\ldots,\boldsymbol{e_{i,j}}\}$ represents edge-set, $\boldsymbol{e_{i,j}}$ represents the edge constructed between the i-th and j-th lncRNA sequence, $\boldsymbol{e_{i,j}}=\boldsymbol{1}$ represents the existence edge, and $\boldsymbol{e_{i,j}}=\boldsymbol{0}$ represents the non-existence edge. $\boldsymbol{X}=\{\boldsymbol{x_1},\boldsymbol{x_2},\ldots,\boldsymbol{x_n}\}$ represents the node features and $\boldsymbol{x_i}$ is the initial feature vector of the node $\boldsymbol{v_i}\in\boldsymbol{V}$ in the graph $\boldsymbol{G}$. Let $\boldsymbol{Y}=\{\boldsymbol{y_1},\boldsymbol{y_2},\ldots,\boldsymbol{y_{|C|}}\}$ indicate label set, which means there was $|\boldsymbol{C}|$ different subcellular location. Our goal is to predict the subcellular location (label) $\boldsymbol{y_i}\in\boldsymbol{Y}$ of a lncRNA ($\boldsymbol{v_i}\in\boldsymbol{V}$) by aggregating the node feature($\boldsymbol{x_i}$) of the lncRNA ($\boldsymbol{v_i}$) and the feature information of its neighbor nodes.

Therefore, the graph consists of three parts, the node-set $\boldsymbol{V}$, the node features $\boldsymbol{X}$ and the edge-set $\boldsymbol{E}$. The construction steps are as follows:

Step 1: To calculate the cosine similarity in **Step 3**, the low-level features are extracted from $\boldsymbol{V}=\{\boldsymbol{v_1},\boldsymbol{v_2},\ldots,\boldsymbol{v_n}\}$ by k-mer, and then mark them as $\boldsymbol{L}=\{\boldsymbol{l_1},\boldsymbol{l_2},\ldots,\boldsymbol{l_n}\}$;
Step 2: To learn the high-level features by the classifier, the low-level features extracted from each lncRNA sequence are expressed as the initial features of the corresponding node, forming the node features $\boldsymbol{X}=\{\boldsymbol{x_1},\boldsymbol{x_2},\ldots,\boldsymbol{x_n}\}$;
Step 3: Calculate the cosine similarity $\boldsymbol{S}$ between the low-level features $\boldsymbol{L}$ from **Step 1**. When the cosine similarity $\boldsymbol{S_{i,j}}$ between two low-level features $\boldsymbol{l_i}$ and $\boldsymbol{l_j}$ is greater than a certain threshold $\boldsymbol{\tau}$, an edge is created for the two nodes($\boldsymbol{e_{i,j}}=\boldsymbol{1}$; otherwise, $\boldsymbol{e_{i,j}}=\boldsymbol{0}$). As shown in eqs. (1) and (2).

$$\boldsymbol{S_{i,j}}=\frac{\boldsymbol{l_i}\bullet\boldsymbol{l_j}}{\|\boldsymbol{l_i}\|\|\boldsymbol{l_j}\|} \tag{1}$$

$$\boldsymbol{e_{i,j}}=\begin{cases}\boldsymbol{1},\boldsymbol{S_{i,j}}\geq\boldsymbol{\tau}\\ \boldsymbol{0},\boldsymbol{S_{i,j}}<\boldsymbol{\tau}\end{cases} \tag{2}$$

$\boldsymbol{\tau}$ is a hyperparameter, which we will discuss further in Section 3.2. It should be noted that different methods can be used to extract low-level features from lncRNA sequences in Step 1 and Step 2. By experiment comparisons, we found that GM-lncLoc performs best when k-mer was used for both similarity features and node features, as shown in Table 2. In addition, the final constructed graph is allowed to have isolated nodes, which implies support for new lncRNA prediction, as shown in Fig. 3.

Table 2 The performance with different features

features of calculating cosine similarity	features of node features	F1	Recall	Acc
k-mer(k = 5)	**k-mer**(k = 5)	**0.833**	**0.835**	**0.822**
	RevKmer(k = 5)	0.713	0.714	0.721
	PseDNC(λ = 150 and ω = 0.3)	0.529	0.530	0.529
k-mer(k = 5)	**k-mer**(k = 5)	**0.833**	**0.835**	**0.822**
RevKmer(k = 5)		0.743	0.742	0.754
PseDNC(λ = 150 and ω = 0.3)		0.662	0.658	0.661

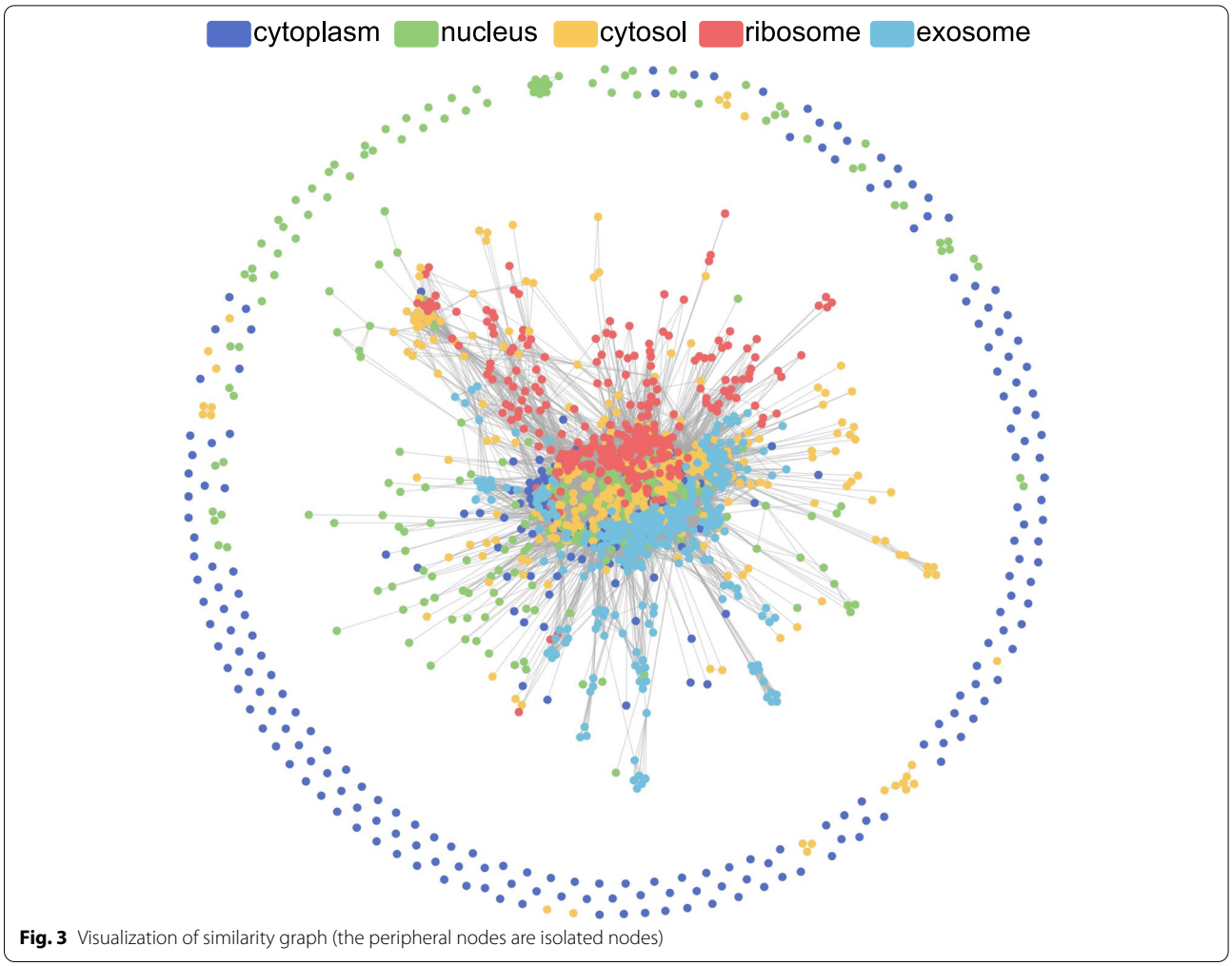

Fig. 3 Visualization of similarity graph (the peripheral nodes are isolated nodes)

GNN based on Meta-learning

Graph Convolutional Network (GCN)

GCN [36] is a semi-supervised learning graph neural network that can be applied to tasks such as node classification and link prediction. The input of GCN consists of two parts: $X_{n\times s}$ and $A_{n\times n}$; where $X_{n\times s}$ represents the n × s feature matrix, while $A_{n\times n}$ represents the n × n adjacency matrix. The output matrix is $Y_{n\times|C|}$, where $|C|$ represents the number of labels, and $Y_{i,j}$ represents the probability that node v_i is predicted to be in the j-th label. The formula of GCN is defined as eq (3).

$$Y = f(X,A) = \sigma\left(D'^{-\frac{1}{2}} A' D'^{-\frac{1}{2}} XW\right) \tag{3}$$

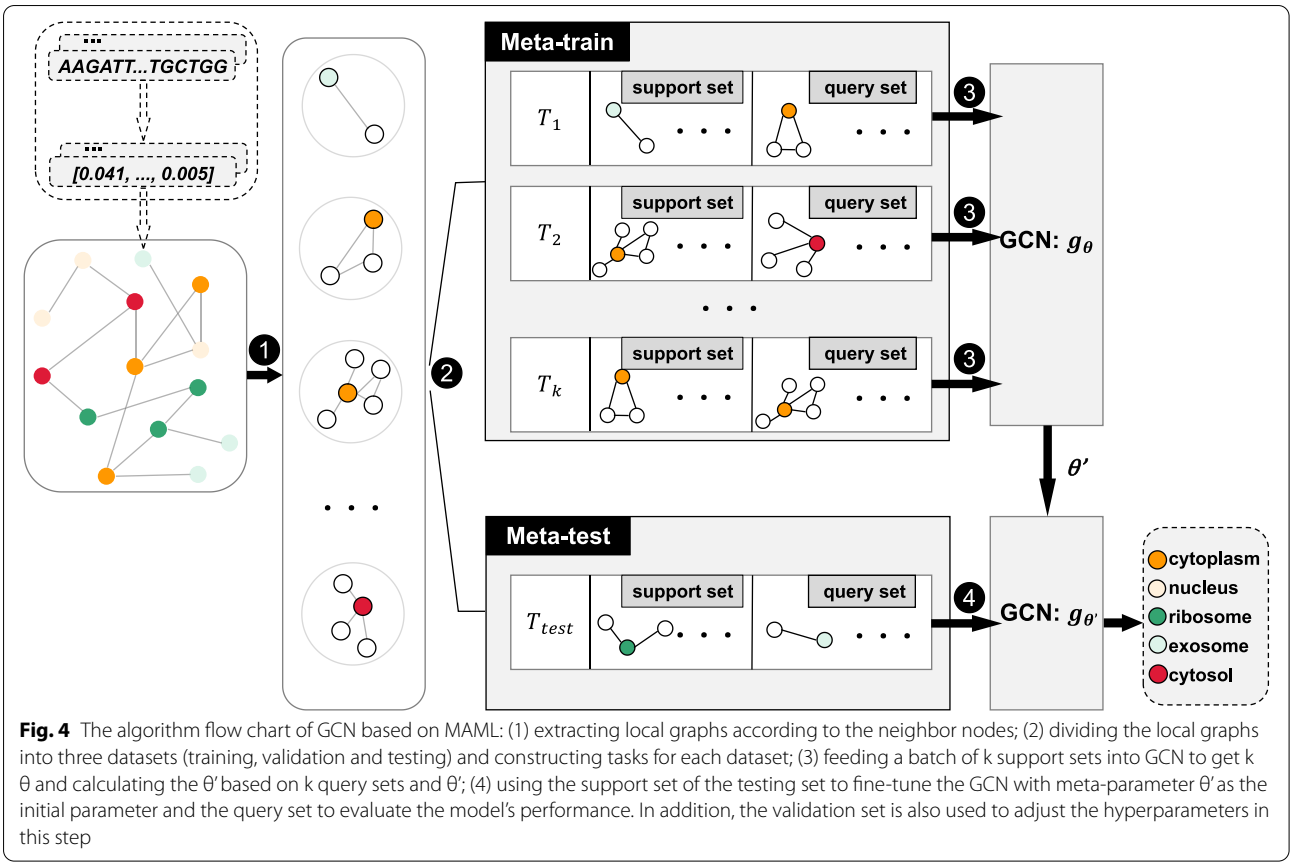

Fig. 4 The algorithm flow chart of GCN based on MAML: (1) extracting local graphs according to the neighbor nodes; (2) dividing the local graphs into three datasets (training, validation and testing) and constructing tasks for each dataset; (3) feeding a batch of k support sets into GCN to get k θ and calculating the θ′ based on k query sets and θ′; (4) using the support set of the testing set to fine-tune the GCN with meta-parameter θ′ as the initial parameter and the query set to evaluate the model's performance. In addition, the validation set is also used to adjust the hyperparameters in this step

where $\boldsymbol{A}^{'}=\boldsymbol{A}+\boldsymbol{E}$, $\boldsymbol{E}$ notes an identity matrix; $\boldsymbol{D}^{'}$ is the ***degree matrix***[3] of $\boldsymbol{A}^{'}$; $\boldsymbol{W}$ is the weight matrix and $\boldsymbol{\sigma}$ notes an activation function.

MAML

MAML [33] is an outstanding model in meta-learning because of its simplicity and universality. Meta-learning focuses on learning meta-knowledge from a series of tasks, so as to learn the parameters of new tasks quickly. In MAML, the set of functions $\{\boldsymbol{g_1}, \boldsymbol{g_2}, \ldots, \boldsymbol{g_k}\}$ in Meta-train learn the meta-parameters $\boldsymbol{\theta}^{'}$ (meta-knowledge) through $\boldsymbol{k}$ tasks $\{\boldsymbol{T_1}, \boldsymbol{T_2}, \ldots, \boldsymbol{T_k}\}$, and then let meta-parameters be used as the initial parameters of the function $\boldsymbol{g}$ in Meta-test to quickly adapt to the new task T. MAML can be simply understood as a training mode: *Pre-training*[4] from Meta-train + Fine-tuning in Meta-test. This training mode can not only effectively deal with the problem of few-shot learning, but also significantly reduce the training time for new tasks employing meta-parameters, which can be verified by the experiment in **Section 3.5**.

GCN based on MAML

The data of lncRNA were transformed into graphical data, while the problem of fewer lncRNA samples still exists. Therefore, we combined GCN and MAML in predicting lncRNA subcellular localization, that is, the training mode of MAML is applied to the training of GCN model. Since the training of MAML is task-based, and tasks need to be constructed by repeatedly sampling from the dataset. To fit the training mode of MAML, local graphs of each node in graph need to be extracted first. The algorithm flow chart is shown in Fig. 4. The details are as follows:

1) Extracting local graph: In **Section 2.2**, we have constructed graph $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E},\boldsymbol{X})$ for lncRNA. Then we extract each node $\{\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_n}\}$ and its neighbor nodes in graph $\boldsymbol{G}$ to form the corresponding local graph $\{\boldsymbol{G_1}, \boldsymbol{G_2}, \ldots, \boldsymbol{G_n}\}$ of $\boldsymbol{n}$ nodes, where $\boldsymbol{G_i} \in \boldsymbol{G}$ represents the local graph of the i-th node, and $\boldsymbol{G_i}=\{\boldsymbol{V_i}, \boldsymbol{E_i}, \boldsymbol{X_i}\}$, $\boldsymbol{V_i}=\{\boldsymbol{v_i}\} \cup \{\boldsymbol{v_j} \in \boldsymbol{V} | \boldsymbol{e_{i,j}}=\boldsymbol{1}\}$,

[3] $D_{i,j} = \begin{cases} \deg(v_i), \text{if } i=j \\ 0, \text{othewise} \end{cases}$, where *deg*($v_i$) notes the degree of the vertex v_i.

[4] Pre-training usually requires a large amount of data, but MAML is originally proposed for few-shot learning, which samples the same sample multiple times when generating a series of tasks in Meta-train [33]. Here it can be understood as a special kind of pre-training.

$E_i = \{e_{i,j} \in E | e_{i,j} = 1\}$, $X_i = \{x_i\} \cup \{x_j \in X | e_{i,j} = 1\}$. Thus, 1460/1668 local graphs (samples) of lncRNA can be obtained, that is $D = \{G_1, G_2, \ldots, G_{1460/1668}\}$;

2) Dividing dataset: Firstly, the dataset $D = \{G_1, G_2, \ldots, G_n\}$ is divided into three data sets: $D_{train} = \{G_a, \ldots, G_o\}$, $D_{val} = \{G_b, \ldots, G_p\}$ and $D_{test} = \{G_c, \ldots, G_q\}$, and the following condition are satisfied $\begin{cases} D_{train} \cap D_{val} \cap D_{test} = \varnothing \\ D_{train} \cup D_{val} \cup D_{test} = D \end{cases}$. Then, according to the MAML method, m tasks $T_{train} = \{T_1, T_2, \ldots, T_m\}$ are composed of randomly selected $|C| \times (k_{support} + k_{query})$ samples G_i repeatedly, where $|C|$ represents the number of location labels, $k_{support}$, k_{query} and m are the hyperparameter; The samples G_i in D_{val} and D_{test} constitute a single task T_{val} and T_{test} respectively; Finally, each task is further divided into support set and query set, denoted as $T_{i-support}$ and $T_{i-query}$, (respectively;)

3) Meta-train: Firstly, m tasks' $T_{train-support}$ of T_{train} are input into m GCNs(i.e. f_θ) with initial parameters θ for training, and m corresponding parameters $\{\theta_1, \theta_2, \ldots, \theta_m\}$ are obtained after updating respectively; Then, the total loss is calculated for updating θ by m tasks' $T_{train-query}$ and $\{f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_m}\}$ in T_{train}. Finally, the optimized meta-parameter θ' is obtained;

4) Meta-test: $T_{test-support}$ of T_{test} is used to fine-tune the GCN(i.e.$f_{\theta'}$) with meta-parameter θ' as the initial parameter, then $T_{test-query}$ is used to evaluate the performance of $f_{\theta'}$. In the actual training, T_{val} is used before the Meta-test in step 4 to verify the model, and then adjust the hyperparameters. Moreover, another graph is constructed by the independent test set, dataset3. Therefore, there are no overlaps between the training data and the independent test dataset.

Performance evaluation

To evaluate the performance of GM-lncLoc, the following evaluations criterion is performed based on 10-fold cross-validation. In addition to the typical Accuracy (Acc), Recall(R) and F_1 Score (F1) are also included. The formula is shown below.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$P^{(i)} = \frac{TP^{(i)}}{TP^{(i)} + FP^{(i)}} \tag{5}$$

$$R^{(i)} = \frac{TP^{(i)}}{TP^{(i)} + FN^{(i)}} \tag{6}$$

$$F1 = \frac{1}{|C|} \sum_{i=1}^{|C|} 2 \times \frac{P^{(i)} \times R^{(i)}}{P^{(i)} + R^{(i)}} \tag{7}$$

$$R = \frac{1}{|C|} \sum_{i=1}^{|C|} R^{(i)} \tag{8}$$

where TP, FP and FN represent true positive, false positive and false negative, respectively. P represents Precision, and $|C|$ represents the number of location labels.

Results and discussion

Performance comparison of different node features

To explore the effect of feature extraction method, we compared the prediction results of three low-level feature extraction methods, including k-mer [39], RevKmer [40, 41] and PseDNC [42–44], which is on the basis of the previous study [14, 25–31] in dataset1. As shown in Table 2, the k values of both k-mer and RevKmer are 5, and the λ and ω of PseDNC are set to 150 and 0.3 respectively. First of all, the low-level features extracted by k-mer are fixed as the features of calculating cosine similarity, and comparing the features extracted by the three methods as node features, then the accuracy of 82.2, 72.1, and 52.9% are obtained, respectively. Next, fixing the low-level features extracted by k-mer as node features and comparing the low-level features extracted by the three methods as the features of calculating cosine similarity, the accuracy of 82.2, 75.4, and 66.1%, are obtained, respectively.

As we can see from the results, GM-lncLoc shows the best performance when the low-level features extracted by k-mer are used in calculating cosine similarity and the node features. RevKmer removes some frequency of base sequences on the basis of k-mer, which is essentially a dimensionality reduction operation and may lose some information; PseDNC is a method based on pseudo dinucleotide composition, which may be limited to dinucleotide and fail to extract more critical information.

Performance comparison of different values of threshold τ

When constructing the graph, the value of threshold τ directly determines the structure of the graph, especially the number of edges. Therefore, it is necessary to discuss

Table 3 The performance with different threshold τ

τ	isolated nodes	edges	key edges(%)	F1	Recall	Acc
0.4	251	90,210	44.7	$0.895_{\pm 0.012}$	$0.895_{\pm 0.012}$	$0.897_{\pm 0.013}$
0.5	324	31,478	73.9	$0.922_{\pm 0.017}$	$0.923_{\pm 0.011}$	$0.924_{\pm 0.012}$
0.6	352	19,942	84.8	$0.928_{\pm 0.013}$	$0.927_{\pm 0.013}$	$0.929_{\pm 0.014}$
0.7	367	13,872	91.3	$\mathbf{0.933_{\pm 0.011}}$	$\mathbf{0.933_{\pm 0.013}}$	$\mathbf{0.934_{\pm 0.01}}$
0.8	408	9072	96.4	$0.915_{\pm 0.015}$	$0.914_{\pm 0.014}$	$0.914_{\pm 0.012}$
0.9	525	4692	99.4	$0.843_{\pm 0.011}$	$0.834_{\pm 0.014}$	$0.834_{\pm 0.012}$

Table 4 The performance with different k values in k-mer

	F1	Recall	Acc	Time(s)
3-mer	$0.558_{\pm 0.016}$	$0.560_{\pm 0.011}$	$0.513_{\pm 0.014}$	5246
4-mer	$0.760_{\pm 0.013}$	$0.761_{\pm 0.013}$	$0.761_{\pm 0.014}$	6250
5-mer	$0.829_{\pm 0.014}$	$0.830_{\pm 0.013}$	$0.831_{\pm 0.015}$	8378
6-mer	$0.897_{\pm 0.013}$	$0.898_{\pm 0.014}$	$0.900_{\pm 0.011}$	23,890
7-mer	$\mathbf{0.933_{\pm 0.011}}$	$\mathbf{0.933_{\pm 0.013}}$	$\mathbf{0.934_{\pm 0.01}}$	48,993

As shown in Table 3, as the value of τ increases, the number of isolated nodes in the graph also increases, while the number of edges decreases. The number of isolated nodes and edges is directly related to the structural information of the graph. Meanwhile, the proportion of key edges and the overall performance of GM-lncLoc are improving as the value of τ increases from 0.4 to 0.7. However, when τ rises from 0.7 to 0.9, although the proportion of key edges also rises from 91.3 to 99.4%, the

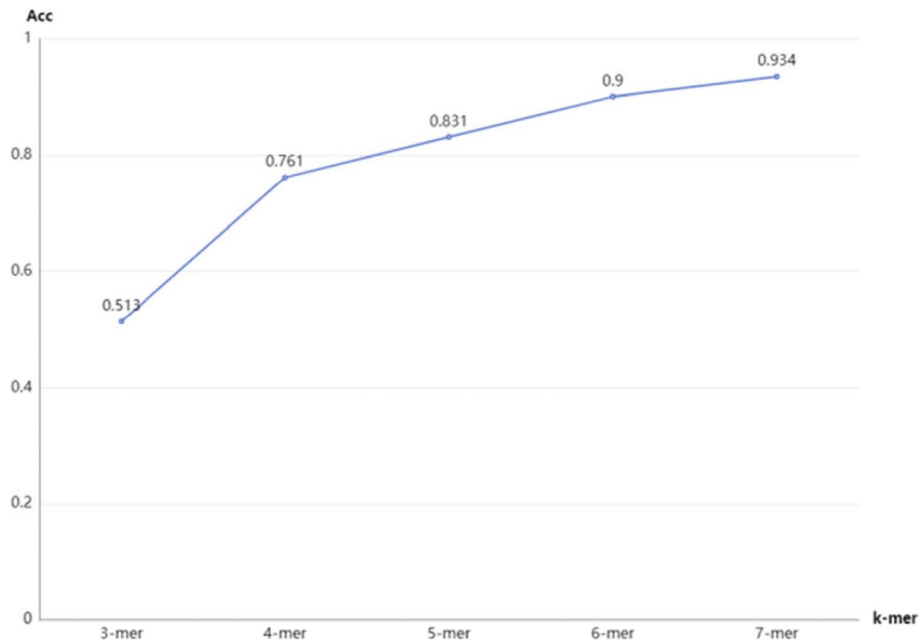

Fig. 5 The accuracy with different k values in k-mer

the influence of threshold τ on GM-lncLoc. While the value of τ is set from 0.4 to 0.9, we compared the performance of GM-lncLoc in dataset1. In addition, to evaluate the impact of edges on the model, we also tallied the number of isolated nodes and edges in the graph, and the proportion of key edges.[5]

[5] For the convenience of expression, we call the edge connecting two nodes of the same class the key edge.

performance of GM-lncLoc deteriorates. It indicates that the performance of GM-lncLoc is not only related to the number of isolated nodes and edges but also related to the proportion of the key edge.

Performance comparison of different k values in k-mer

To explore the effect of k value on our model, the k value is set from 3 to 7. The comparative experiment in dataset1

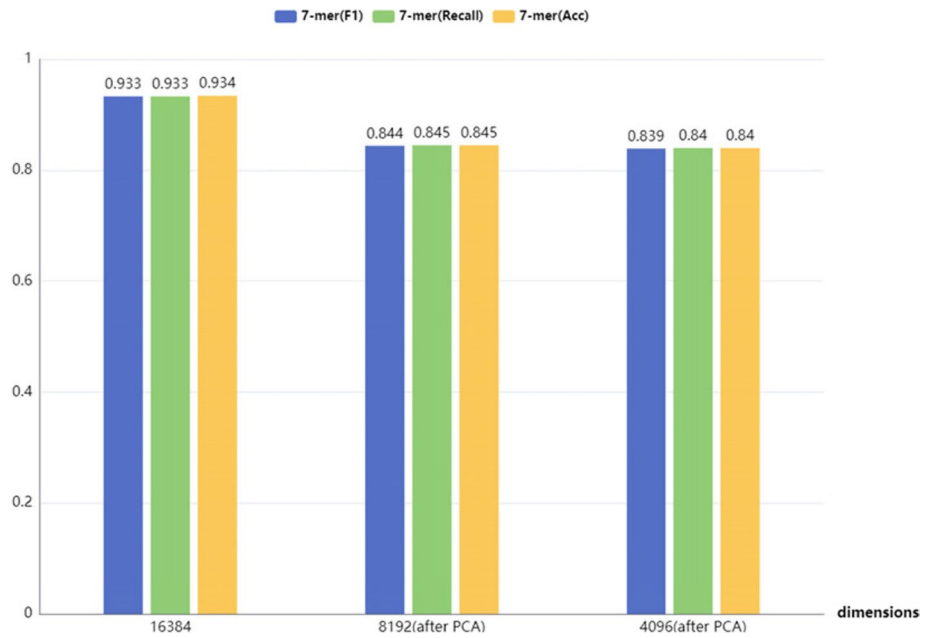

Fig. 6 The performance after dimension reduction of features from 7-mer

is shown in Table 4**.** and Fig. 5. Since the dimension of the 7-mer frequency vector is 16,384, the higher k-mer frequency feature is not conducted in our experiments considering the time cost and equipment conditions. It can be apparently seen from Table 4**.** that the performance of GM-lncLoc improves as the increase of k value.

In addition, the dimension may be too high when the feature is the 7-mer frequency vector. Therefore, we also tried to utilize the PCA algorithm to reduce the dimension of the node features from 16,384 to 8192 and 4096, and then compared their performance. It can be seen in Fig. 6 that the accuracy after dimension reduction has not been improved, but rather decreased. We believe that the loss of some information after the dimensionality reduction operation is what makes it ineffective.

Performance comparison of different number of neighbor node's layer

GCN aggregates information about neighbor nodes during computing node embedding. In this paper, if there is an edge between node A and node B, node B is called one of the first-layer neighbor nodes of node A. Furthermore, if there is also an edge between node B and node C, node C is called one of the second-layer neighbor nodes of node A. Thus, when the neighbor node information is aggregated, there is a difference between the neighbor node information of the first layer and that of the first two layers. We conducted a relevant experimental comparison in dataset1, and the results are shown in Fig. 7. It can be seen that the model performance on the first-layer neighbor aggregation is slightly higher than that of the first two layers. However, the latter consumes 2 to 3 times as much memory than the former in terms of memory consumption. As a result, the first-layer neighbor aggregation is adopted in our experiments.

Among them, memory consumption is in line with our intuitive understanding. The number of neighbor nodes at the first two layers must be larger than that of the neighbor nodes at the first layer, so more memory is required. In addition, the accuracy of the neighbor node information of the first layer of aggregation is higher than that of the neighbor node information of the first two layers of aggregation, which is also consistent with the theoretical proof in Kexin Huang [35] et al., which proves that the interaction between two nodes decreases exponentially as their distance increases. In other word, as the distance increases, the number of neighbor nodes increases exponentially, while the information provided by neighbor nodes decreases exponentially. Therefore, the further the distance, the less efficient the information aggregation is.

Performance comparison of GM-lncLoc and GCN

To validate the effectiveness of MAML, we trained GCN alone in dataset1. As shown in Table 5, the results of GCN alone for predicting lncRNA subcellular localization are not good due to the limited amount of data. On the contrary, GM-lncLoc is able to predict lncRNA subcellular localization more effectively with about 0.4

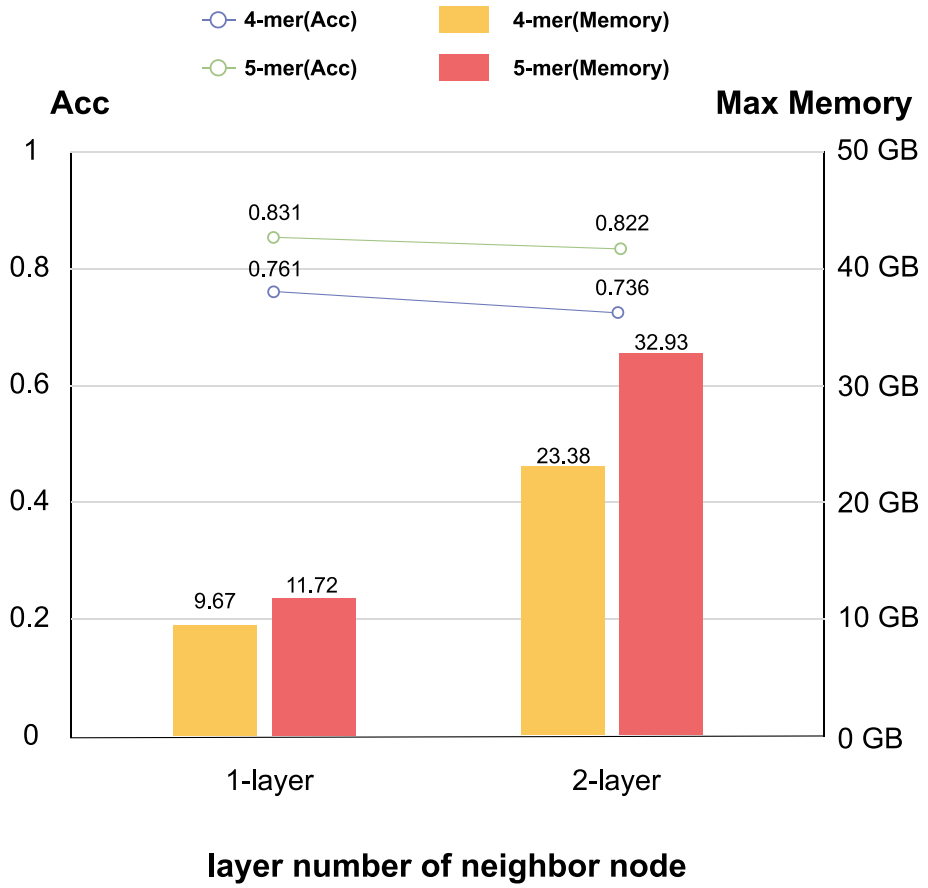

Fig. 7 The performance with different number of neighbor node's layer

Table 5 The performance comparison of GCN and GM-lncLoc (Ours)

	F1	Recall	Acc
GCN	–	–	0.531
GM-lncLoc (Ours)	**0.933**	**0.933**	**0.934**

higher accuracy than GCN alone. In the experiment process, we also find that it only takes about 34.4 seconds to complete the training using meta-parameters as the initial parameters of the meta-test task, while it takes about 325.3 seconds for GCN to complete the training. From the perspective of training duration, GCN took nearly 9.5 times longer than training with meta-parameters, which indicates that the meta-parameters obtained by GM-lncLoc can significantly improve the training efficiency.

Performance comparison with other methods

In this section, we utilize the method of 10-fold cross-validation to compare the GM-lncLoc with previous methods, as shown in Table 6 and Table 7.[6] It is evident that GM-lncLoc has achieved the best results both on the dataset1 and dataset2. In the dataset1, the accuracy of GM-lncLoc is about 34.3% higher than lncLocator [25]; In the dataset2, the accuracy of GM-lncLoc is about 1.8% higher than the current highest LncLocPred [30]. It demonstrates the superiority of our proposed GM-lncLoc in lncRNA subcellular localization prediction. In

Table 6 Comparison with existing state-of-the-art methods (dataset1 with 5 subcellular compartments)

Method	F1	Recall	Acc(%)
lncLocator [25]	0.367	0.363	59.1
DeepLncLoc [31]	0.563	0.524	53.7
GM-lncLoc (Ours)	0.933	0.933	93.4
GM-lncLoc (the test set consists of real samples)	0.901	0.902	90.3

[6] In order to facilitate comparison with the methods with dataset2, we introduce other evaluations in Table 7, including Sensitivity, Specificity and MCC.

Table 7 Comparison with existing state-of-the-art methods (dataset2 with 4 subcellular compartments)

Method	Location	Sensitivity(%)	Specificity(%)	MCC	Acc(%)
iLoc-lncRNA [14]	Cytoplasm	99.06	67.68	0.742	86.72
	Nucleus	77.56	97.59	0.796	
	Ribosome	46.51	99.83	0.652	
	Exosome	16.67	1.00	0.400	
Locate-R [28]	Cytoplasm	84.74	89.10	0.725	90.69
	Nucleus	65.92	95.15	0.658	
	Ribosome	100.00	98.37	0.970	
	Exosome	100.00	99.17	0.978	
LncLocPred [30]	Cytoplasm	99.10	85.60	0.876	92.37
	Nucleus	96.80	96.80	0.915	
	Ribosome	60.50	99.80	0.751	
	Exosome	20.00	100.00	0.439	
Proposed by Yang et al. [29]	Cytoplasm	100.00	84.14	0.880	90.37
	Nucleus	82.47	97.14	0.821	
	Ribosome	41.86	99.83	0.615	
	Exosome	66.67	98.21	0.639	
GM-lncLoc (Ours)	Cytoplasm	93.21	96.06	0.879	94.20
	Nucleus	88.85	98.21	0.889	
	Ribosome	96.80	98.99	0.959	
	Exosome	99.07	99.38	0.982	
GM-lncLoc (the test set consists of real samples)	Cytoplasm	99.00	92.37	0.860	93.00
	Nucleus	73.80	99.47	0.811	
	Ribosome	99.20	99.83	0.991	
	Exosome	100.00	99.00	0.980	

particular, the samples are more imbalanced in the dataset1, and our method provides a significant improvement over existing methods. This shows that our method is more advantageous in the case of an imbalanced sample. To improve the persuasion, we set all the samples in the test set as real samples in dataset1 and dataset2 and obtained an accuracy of 90.3 and 93.1%, respectively. Although the accuracy is slightly lower than that of the 10-fold cross-validation method, it is still better than other methods. Moreover, we compare GM-lncLoc with the three methods, iLoc-lncRNA, Locate-R and LncLocPred, in the independent test set (dataset3), and GM-lncLoc attains a better accuracy, 46.21%. Besides, F1 and Recall are 0.469 and 0.463, respectively. However, LncLocPred [25] had not provided other performance evaluations in iLoc-lncRNA, Locate-R and LncLocPred. As shown in Table 8, the result indicates that our model does not depend on a particular dataset, which is better in generalization. To provide strong support to the research, we describe the algorithms and features used for each method in Table 9.

On the one hand, GM-lncLoc is based on GNN and is able to extract high-level features from low-level features of lncRNA sequences to complete classification tasks, while traditional machine learning methods complete classification based on low-level features of lncRNA sequences; On the other hand, GM-lncLoc extract correlation information between lncRNAs based on sequence information, which is unable to be achieved by previous methods. The comparison experiments between GM-lncLoc and previous methods, especially lncLocator, demonstrate the significance of graph structure information for GM-lncLoc.

Table 8 Comparison with other methods on the independent dataset (dataset3)

Method	Acc(%)
iLoc-lncRNA [14]	35.86
Locate-R [28]	38.64
LncLocPred [30]	44.44
GM-lncLoc (Ours)	46.21

Table 9 Description of the algorithms and features used for each method

Method	Feature	Oversampling	Algorithm	The number of subcellular compartments
DeepLncRNA [26]	k-mer, Genome loci, RNA binding motifs	–	Neural networks	2
lncLocator 2.0 [27]	–	–	CNN, LSTM, Multi-layer perceptron	2
iLoc-lncRNA [14]	PseKNC	–	SVM	4
Locate-R [28]	k-mer, n-gapped k-mer	SMOTE	Locally Deep SVM	4
LncLocPred [30]	k-mer, PseDNC, Triplet	–	Logistic regression	4
lncLocator [25]	k-mer	SOS [55]	Random forest, SVM, Neural networks	5
DeepLncLoc [31]	k-mer	–	TextCNN	5
GM-lncLoc **(Ours)**	k-mer	SMOTE	GCN based on MAML	4 or 5

lncLocator is also based only on the low-level features k-mer and utilizes the over-sampling method to augment the dataset. However, our GM-lncLoc obtains 93.4% accuracy based on the graph, while the accuracy of lncLocator only achieves 59.1%. In addition, our model also utilizes a few-shot training model, so that better results can be obtained in lncRNA subcellular localization problems with a limited number of samples.

Conclusion

In conclusion, our proposed GM-lncLoc based on the combination of Graph Neural Network and Meta-learning is a totally new method for lncRNA subcellular localization prediction. On the one hand, a graph is constructed for the initial data, which is not used in the previous approach; On the other hand, Graph Neural Network and Meta-learning are modeled jointly, which is able to effectively predict lncRNA subcellular localization with only a small number of samples, and obtain the meta-parameters for quickly learning of new lncRNA subcellular localization tasks. The experimental results from lncRNA subcellular localization prediction demonstrate that GM-lncLoc is effective and promising. Even more important, the advantages of GM-lncLoc will become more evident with the addition of new data in the lncRNA database due to the generalization ability of meta-parameters. We have reason to believe that GM-lncLoc can greatly contribute to the further study of lncRNA functional mechanisms in biology.

Acknowledgements
Not applicable.

Authors' contributions
LL and JC conception and design of study, final revision of the manuscript; JC and TW, Writing – original draft, software, formal analysis; XD, data curation, resources; LT, data curation, validation. The author(s) read and approved the final manuscript.

Funding
This research was funded by the National Natural Science Foundation of China (No. 61862067),the Applied Basic Research Project in Yunnan Province (No.202201AT070042) and the NSFC-Yunnan Union Key Grant (No.U1902201).

Availability of data and materials
The datasets and source code are available at https://github.com/JunzheCai/GM-lncLoc.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]School of Information, Yunnan Normal University, Kunming, Yunnan, China. [2]Key Laboratory of Educational Information for Nationalities Ministry of Education, Yunnan Normal University, Kunming, Yunnan, China.

Received: 29 June 2022 Accepted: 21 November 2022
Published online: 28 January 2023

References

1. Chen X, You ZH, Yan GY, et al. IRWRLDA: improved random walk with restart for lncRNA-disease association prediction. Oncotarget. 2016;7(36):57919.
2. Dhanoa JK, Sethi RS, Verma R, et al. Long non-coding RNA: its evolutionary relics and biological implications in mammals: a review. J Anim Sci Technol. 2018;60(1):25.
3. Struhl K. Transcriptional noise and the fidelity of initiation by RNA polymerase II. Nat Struct Mol Biol. 2007;14:103.
4. Gupta RA, Shah N, Wang KC, Kim J, Horlings HM, Wong DJ, et al. Long non-coding rna hotair reprograms chromatin state to promote cancer metastasis. Nature. 2010;464(7291):1071.
5. Johnson R. Long non-coding rnas in huntington's disease neurodegeneration. Neurobiol Dis. 2012;46(2):245–54.
6. Lin R, Maeda S, Liu CA, Karin M, Edgington T. A large noncoding rna is a marker for murine hepatocellular carcinomas and a spectrum of human carcinomas. Oncogene. 2007;26(6):851.
7. McPherson R, Pertsemlidis A, Kavaslar N, Stewart A, Roberts R, Cox DR, et al. A common allele on chromosome 9 associated with coronary heart disease. Science. 2007;316(5830):1488–91.

8. Mourtada-Maarabouni M, Pickard M, Hedge V, Farzaneh F, Williams G. Gas5, a non-protein-coding rna, controls apoptosis and is downregulated in breast cancer. Oncogene. 2009;28(2):195.
9. Panzitt K, Tschernatsch MM, Guelly C, Moustafa T, Stradner M, Strohmaier HM, et al. Characterization of hulc, a novel gene with striking up-regulation in hepatocellular carcinoma, as noncoding rna. Gastroenterology. 2007;132(1):330–42.
10. Pasmant E, Laurendeau I, Héron D, Vidaud M, Vidaud D, Bieche I. Characterization of a germ-line deletion, including the entire ink4/arf locus, in a melanoma-neural system tumor family: identification of anril, an antisense noncoding rna whose expression coclusters with arf. Cancer Res. 2007;67(8):3963–9.
11. Wang J, Liu X, Wu H, Ni P, Gu Z, Qiao Y, et al. Creb upregulates long non-coding rna, hulc expression through interaction with microrna-372 in liver cancer. Nucleic Acids Res. 2010;38(16):5366–83.
12. Zhang X, Rice K, Wang Y, Chen W, Zhong Y, Nakayama Y, et al. Maternally expressed gene 3 (meg3) noncoding ribonucleic acid: isoform structure, expression, and functions. Endocrinology. 2009;151(3):939–47.
13. Zhao J, Dahle D, Zhou Y, Zhang X, Klibanski A. Hypermethylation of the promoter region is associated with the loss of meg3 gene expression in human pituitary tumors. J Clin Endocrinol Metab. 2005;90(4):2179–86.
14. Su ZD, Yan H, Zhang ZY, et al. iLoc-lncRNA: predict the subcellular location of lncRNAs by incorporating octamer composition into general PseKNC. Bioinformatics. 2018;24:24.
15. Donnelly CJ, Fainzilber M, Twiss JL. Subcellular communication through rna transport and localized protein synthesis. Traffic. 2010;11(12):1498–505.
16. Weil TT, Parton RM, Davis I. Making the message clear: visualizing mRNA localization. Trends Cell Biol. 2010;20(7):380–90.
17. Zhang T, Tan P, Wang L, et al. RNALocate: a resource for RNA subcellular localizations. Nuclc Acids Res. 2017;D1:D1.
18. Mas-Ponte D, Carlevaro-Fita J, Palumbo E, Pulido TH, Guigo R, Johnson R. LncATLAS database for subcellular localization of long noncoding RNAs. Rna. 2017;23(7):1080–7.
19. Xiao W, Lin G, Guo X, et al. LncSLdb: a resource for long non-coding RNA subcellular localization. Database. 2018;2018:bay085. https://doi.org/10.1093/database/bay085.
20. Pierleoni A, et al. MemLoci: predicting subcellular localization of membrane proteins in eukaryotes. Bioinformatics. 2011;27:1224–30.
21. Shen H, Chou K. Hum-mPLoc: an ensemble classifier for large-scale human protein subcellular location prediction by incorporating samples with multiple sites. Biochem Biophys Res Commun. 2007;355:1006–11.
22. Shen H, Chou K. A top-down approach to enhance the power of predicting human protein subcellular localization: hum-mPLoc 2.0. Anal Biochem. 2009;394:269–74.
23. Wan S, et al. FUEL-mLoc: feature-unified prediction and explanation of multi-localization of cellular proteins in multiple organisms. Bioinformatics. 2017;33:749–50.
24. Zhou H, et al. Hum-mPLoc 3.0: prediction enhancement of human protein subcellular localization through modeling the hidden correlations of gene ontology and functional domain features. Bioinformatics. 2017;33:843–53.
25. Cao Z, Pan X, Yang Y, Huang Y, Shen HB. The lncLocator: a subcellular localization predictor for long non-coding RNAs based on a stacked ensemble classifier. Bioinformatics. 2018;34(13):2185–94. https://doi.org/10.1093/bioinformatics/bty085.
26. Gudenas BL, Wang L. Prediction of LncRNA Subcellular Localization with Deep Learning from Sequence Features. Sci Rep. 2018;8:16385. https://doi.org/10.1038/s41598-018-34708-w.
27. Lin Y, Pan X. Hong-Bin Shen, lncLocator 2.0: a cell-line-specific subcellular localization predictor for long non-coding RNAs with interpretable deep learning. Bioinformatics. 2021;37(16):2308–16.
28. Aa A, Hao LB, Ss A. Locate-R: Subcellular localization of long non-coding RNAs using nucleotide compositions. Genomics. 2020;112(3):2583–9.
29. Yang X-F, Zhou Y-K, Zhang L, Gao Y, Du P-F. Predicting LncRNA Subcellular Localization Using Unbalanced Pseudo-k Nucleotide Composition. Curr Bioinforma. 2020;15(6). https://doi.org/10.2174/1574893614666190902151038.
30. Fan Y, Chen M, Zhu Q. LncLocPred: Predicting LncRNA Subcellular Localization Using Multiple Sequence Feature Information. IEEE Access. 2020;8:124702–11. https://doi.org/10.1109/ACCESS.2020.3007317.
31. Zeng M, Wu Y, Lu C, et al. DeepLncLoc: a deep learning framework for long non-coding RNA subcellular localization prediction based on subsequence embedding. Brief Bioinform. 2022(1):23.
32. Scarselli F, Gori M, Tsoi AC, et al. The Graph Neural Network Model. IEEE Trans Neural Netw. 2009;20(1):61.
33. Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17): JMLR.org; 2017. p. 1126–35.
34. Nichol A, Schulman J. Reptile: a scalable metalearning algorithm; 2018.
35. Huang K, Zitnik M. Graph meta learning via local subgraphs: NeurIPS; 2020.
36. Kip FTN, Welling M. Semi-Supervised Classification with Graph Convolutional Networks; 2016.
37. Goff LA, Rinn JL. Linking RNA biology to lncRNAs. Genome Res. 2015;25:1456–65. https://doi.org/10.1101/gr.191122.115.
38. Yan K, Arfat Y, Li D, Zhao F, Chen Z, Yin C, Sun Y, Hu L, Yang T, Qian A. Structure Prediction: New Insights into Decrypting Long Noncoding RNAs. Int J Mol Sci. 2016;17(1):132. https://doi.org/10.3390/ijms17010132.
39. Ghandi M, Mohammad-Noori M, Beer MA. Robust kk-mer frequency estimation using gapped kk-mers. J Math Biol. 2014;69:469–500. https://doi.org/10.1007/s00285-013-0705-3.
40. Stafford NW, Scott K, Robert T, et al. Predicting the in vivo signature of human gene regulatory sequences. Bioinformatics. 2005;suppl_1:i338.
41. Gupta S, Dennis J, Thurman RE, et al. Predicting human nucleosome occupancy from primary sequence. PLoS Comput Biol. 2008;4:e1000134.
42. Tan KK, Le Y, Chua MC. Ensemble of deep recurrent neural networks for identifying enhancers via dinucleotide physicochemical properties. Cells. 2019;8(7):767.
43. Fang T, Zhang Z, Sun R, Zhu L, He J, Huang B, et al. RNAm5CPred: Prediction of RNA 5-Methylcytosine sites based on three different kinds of nucleotide composition. Mol Ther Nucleic Acids. 2019;18:739–47.
44. Zhang S, Chang M, Zhou Z, Dai X, Xu Z. PDHS-ELM: Computational predictor for plant DNase I hypersensitive sites based on extreme learning machines. Mol Gen Genomics. 2018;293(4):1035–49.
45. Zhu PP, Li WC, Zhong ZJ, Deng EZ, Ding H, Chen W, et al. Predicting the subcellular localization of mycobacterial proteins by incorporating the optimal tripeptide into the general form of pseudo amino acid composition. Mol BioSyst. 2015;11:558–63.
46. Zhao YW, Su ZD, Yang W, Lin H, Chen W, Tang H. IonchanPred2.0: a tool to predict ion channels and their types. Int J Mol Sci. 2017;18:1838.
47. Chen W, Yang H, Feng P, Ding H, Lin H. iDNA4mC: identifying DNA N4-methylcytosine sites based on nucleotide chemical properties. Bioinformatics. 2017;33:3518–23.
48. Feng P, Yang H, Ding H, Lin H, Chen W, Chou KC. iDNA6mA-PseKNC: identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. Genomics. 2019;111:96–1002.
49. Yang J, Richard J, Zhang Y, et al. High-accuracy prediction of transmembrane inter-helix contacts and application to GPCR 3D structure modeling. Bioinformatics. 2013;20:2579–87.
50. Yu DJ, Hu J, Yan H, et al. Enhancing protein-vitamin binding residues prediction by multiple heterogeneous subspace SVMs ensemble. Bmc Bioinformatics. 2014;15:297. https://doi.org/10.1186/1471-2105-15-297.
51. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321–57.
52. Atkinson HJ, Morris JH, Ferrin TE, et al. Using Sequence Similarity Networks for Visualization of Relationships Across Diverse Protein Superfamilies. PLoS One. 2009;4(2):e4345.
53. Bouvier, Jason, T, et al. Enzyme Function Initiative-Enzyme Similarity Tool (EFI-EST): A web tool for generating protein sequence similarity networks. Biochimica et biophysica acta. 2015, 1854(8):1019–1037.
54. Kandlinger F, Plach MG, Merkl R. AGeNNT: annotation of enzyme families by means of refined neighborhood networks. BMC Bioinformatics. 2017;18:274. https://doi.org/10.1186/s12859-017-1689-6.
55. Hu J, He X, Yu DJ, et al. A New Supervised Over-Sampling Algorithm with Application to Protein-Nucleotide Binding Residue Prediction. PLoS One. 2014;9(9):e107676.

Publisher's Note